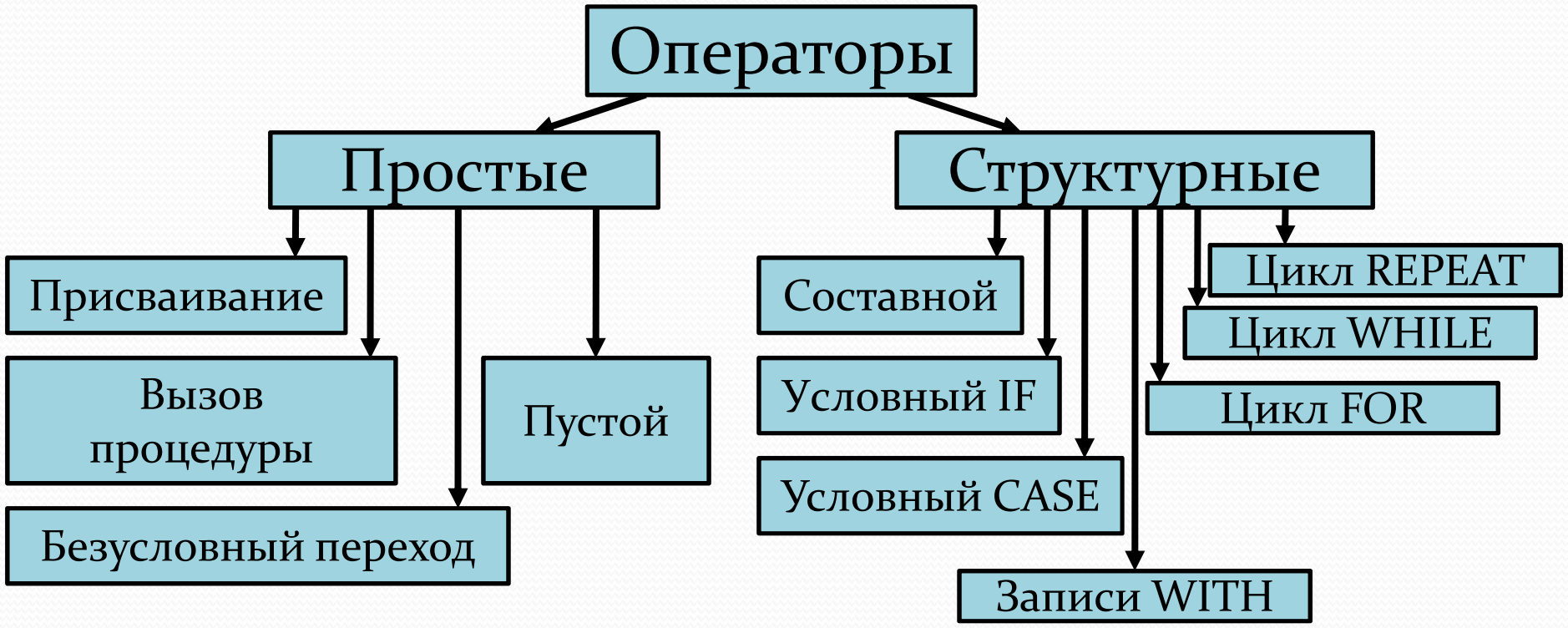
Программирование на языке Pascal. Простые и структурные операторы.

Операторы

Операторы предназначены для описания действий, которые будут выполнены при реализации алгоритма. Операторы отделяются друг от друга символом "точка с запятой" ;



ПРОСТЫЕ ОПЕРАТОРЫ

Оператор присваивания

Состоит из:

- идентификатора переменной
- символа присваивания ": ="
- выражения

Переменная := *выражение* ;

Выполнение оператора присваивания приводит к вычислению значения, определяемого выражением, и присваиванию этого значения переменной, стоящей слева от символа присваивания.

Ограничения оператора присваивания:

- если переменная слева вещественного типа, то арифметическое выражение может быть как целого, так и вещественного типа, то есть содержать либо целые переменные и допустимые для них операторы, либо вещественные, либо те и другие (тогда выражение преобразуется к вещественному типу);
- если переменная слева целого типа, то арифметическое выражение должно быть целочисленным.

Оператор присваивания

$Z := x + y ;$

$A=3 \quad b=5$

$A := b;$

$A=5 \quad b=5$

$b := A;$

$A=3 \quad b=3$

S:= 'Иван Иваныч'

Оператор вызова процедуры

Состоит из:

- Идентификатора
- Списка фактических параметров в круглых скобках

Имя процедуры (*параметры*) ;

Оператор процедуры предназначен для вызова стандартной подпрограммы или подпрограммы, написанной пользователем.

Процедура ввода данных

Процедура ввода помещает вводимое значение переменной в отведенную для нее ячейку.

Read (список ввода);

Эта процедура приостанавливает работу программы и ждет, когда пользователь введет с клавиатуры данные и нажмет клавишу **Enter.**

Read (p);
Read (x,y);

Процедура ввода данных

Процедура ***readln*** отличается от ***read*** тем, что после считывания очередного значения с клавиатуры и присваивания его переменной курсор переходит в начало следующей строки.

Readln (список ввода);

Readln (x);

Процедура вывода данных

Процедура ***write*** предназначена для вывода на экран сообщений и значений переменных.

После слова ***write*** в скобках задается список имен переменных. Кроме имен переменных в список можно включить сообщение — текст, заключенный в апострофы.

Write (список вывода);

Write (‘ текст ‘);

Writeln (‘Целое a = ', a:5);

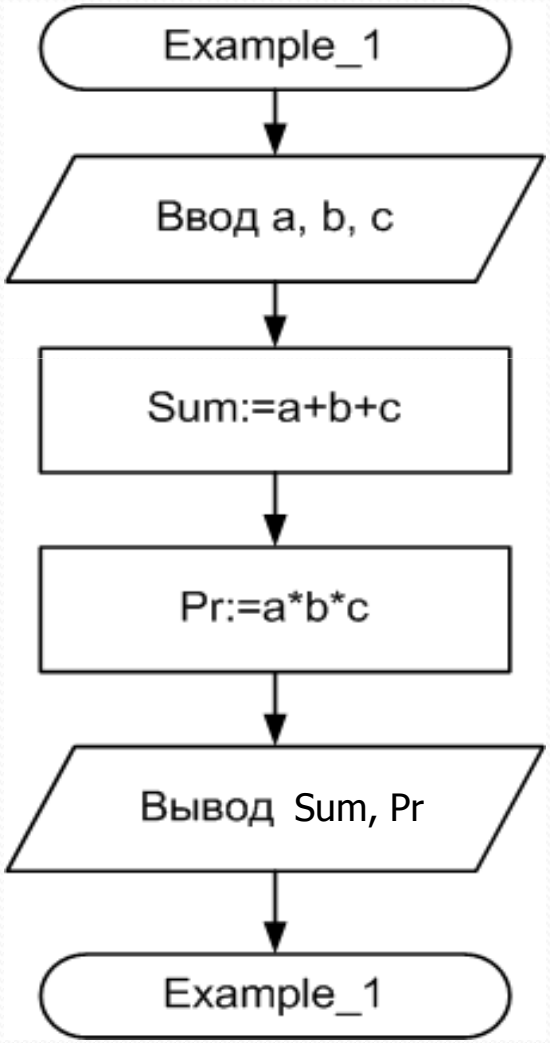
Writeln (‘Вещественное b = ', b:5:2); **31.25**

writeln(′Введите значение переменной X =′);

readln(X);

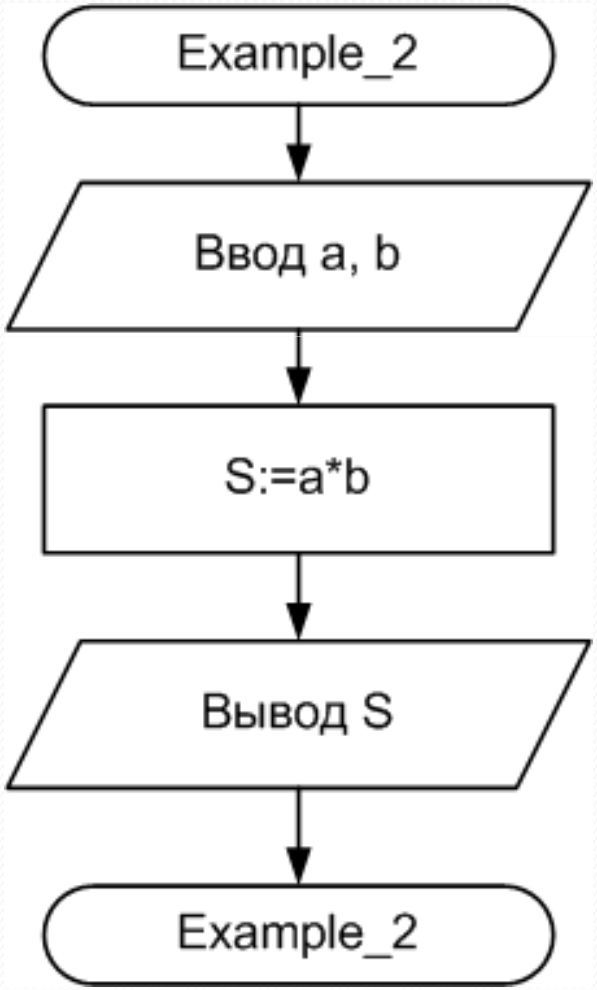
write(′X=′, X:3);

Пример №1 – Найти сумму и произведение трех введенных с клавиатуры целых чисел.



```
Program Example_1;
Uses crt;          {или wincrt}
var a,b,c,sum,pr: integer;
begin
clrscr;
writeln('Программа вычисления суммы и произведения');
write('Введите 3 целых числа через пробел: ');
readln(a,b,c);
sum:=a+b+c;
pr:=a*b*c;
writeln('Сумма = ', sum ,', произведение = ', pr);
writeln('Программа завершена. Нажмите Enter.');
readln
end.
```

Пример №2 – Найти площадь прямоугольника по известным его сторонам.



```
Program Example_2;
Uses crt;        {wincrt}
var a,b,s: real;
begin
clrscr;
writeln('Программа расчета площади прямоугольника');
write('Введите стороны прямоугольника a, b: ');
readln(a,b);
S:=a*b;
write('для сторон A = ', A:5:2 ,', B = ', B:5:2,' - ');
writeln(' площадь прямоугольника S = ', S:7:3);
writeln(' Программа завершена. Нажмите Enter.');
readln
end.
```

Оператор безусловного перехода GOTO

применяется в случаях, когда после выполнения некоторого оператора надо выполнить не следующий по порядку, а какой-либо другой, отмеченный меткой, оператор

go to <метка>;

Метка объявляется в разделе описания меток и состоит из имени и следующего за ним двоеточия.

Имя метки может содержать цифровые и буквенные символы, максимальная длина имени ограничена 127 знаками.

Пример.

```
Program primer;
Label 999, metka;
Begin
….
Go to 999;
…
999: write (‘ Имя’);
…
Go to metka;
….
metka:
write(‘Фамилия’);
…
end.
```

Пустой оператор

не содержит никаких символов и не выполняет никаких действий

Используется для организации перехода к концу блока в случаях, если необходимо пропустить несколько операторов, но не выходить из блока

Пример.

```
Label m;
…
begin
…
go to m;
…
m:
end;
```

СТРУКТУРНЫЕ ОПЕРАТОРЫ

Составной оператор

Составной оператор объединяет группу операторов в единое целое, после чего они считаются одним оператором.

Составной оператор состоит из последовательности объединяемых операторов, которые располагаются между ключевыми словами ***begin*** и ***end***.

Описание оператора	**Пример:**

Begin
оператор1 ;
оператор2;
….
….
оператор n;
end;
	If a>b then
	begin
	a:=2*a;
	b:=3;
	end;

Составной оператор

Условный оператор IF

выражение логического типа (**boolean**), которое может принимать одно из двух значений:

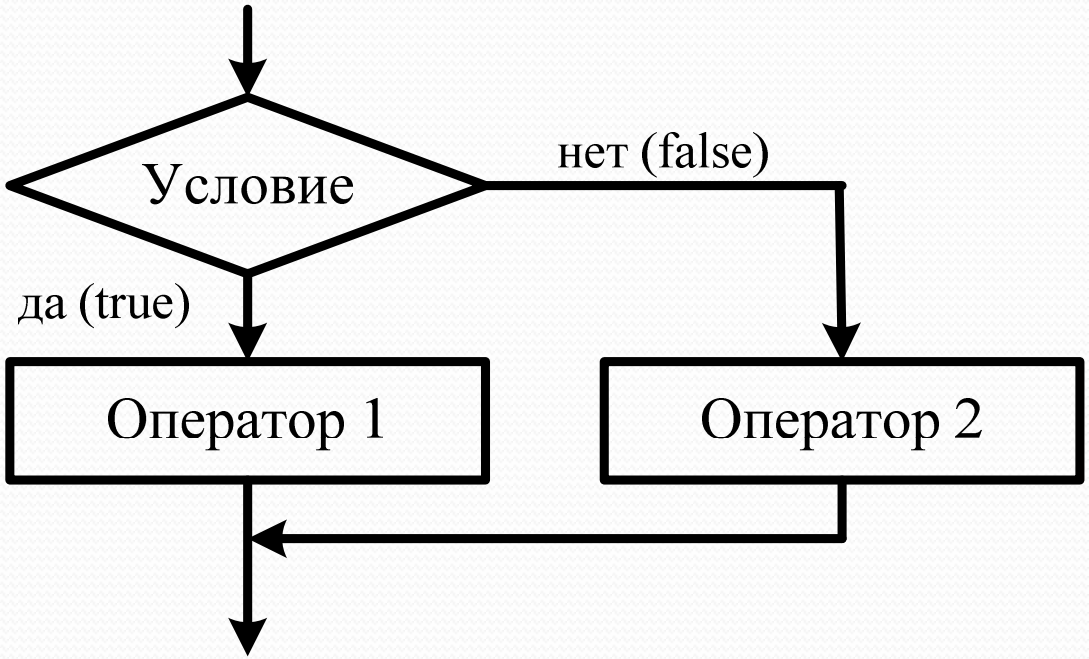
- истина (**true**)
- ложь (**false**)

if *Условие* **then** *Оператор_1* **else** *Оператор_2* ;

Неполное ветвление:

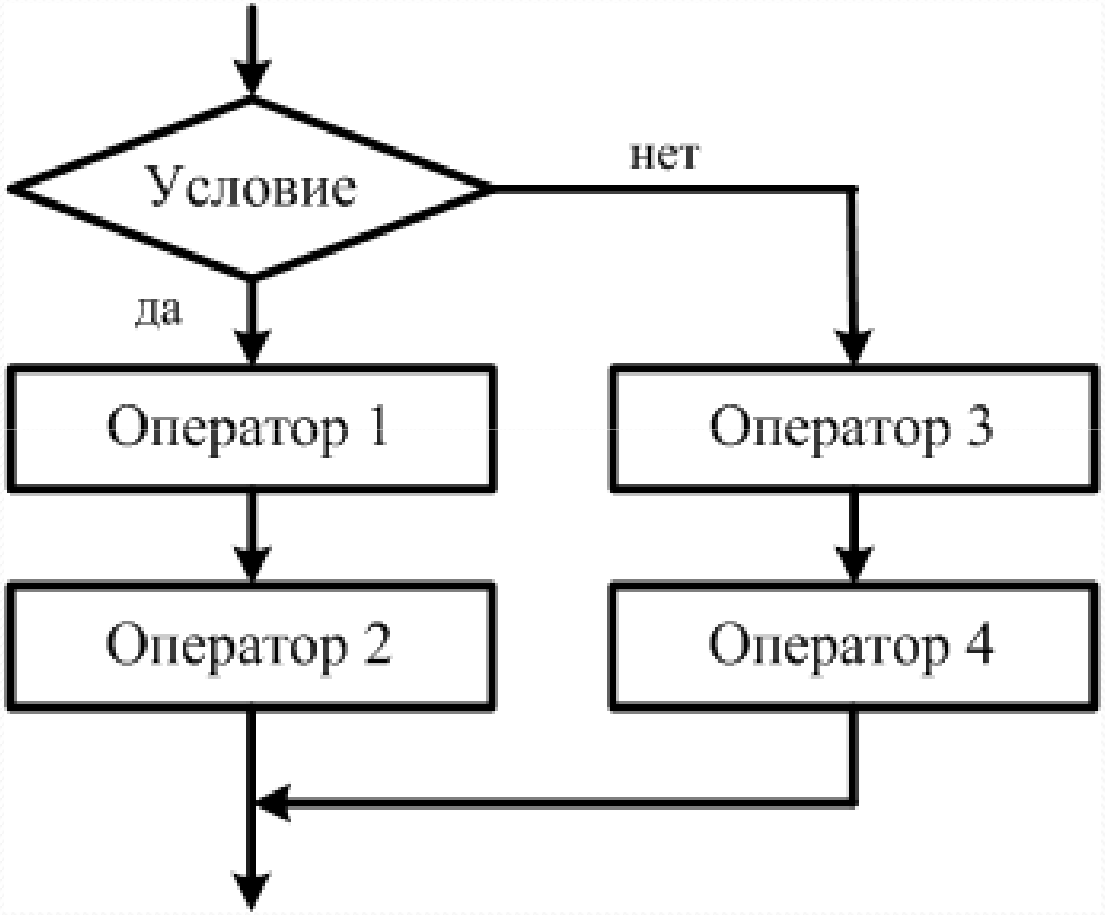
if *Условие* **then** *Оператор_1* ;

Полное ветвление



if < Условие > **then** < Оператор 1 > **else** < Оператор 2 >;

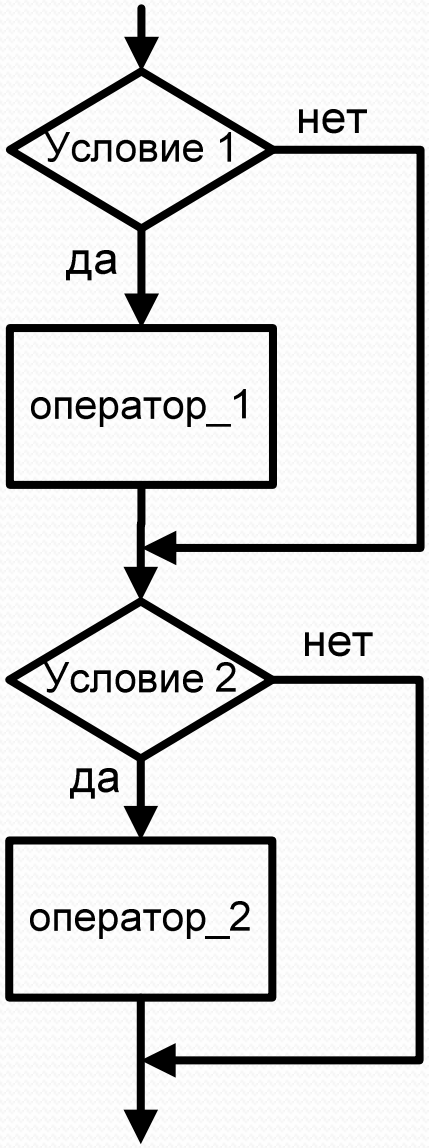
```
if < Условие > then
   Begin
      < Оператор 1 >;
      < Оператор 2 > ;
   end
else
   Begin
      < Оператор 3 >;
      < Оператор 4 > ;
end
```



Неполное ветвление

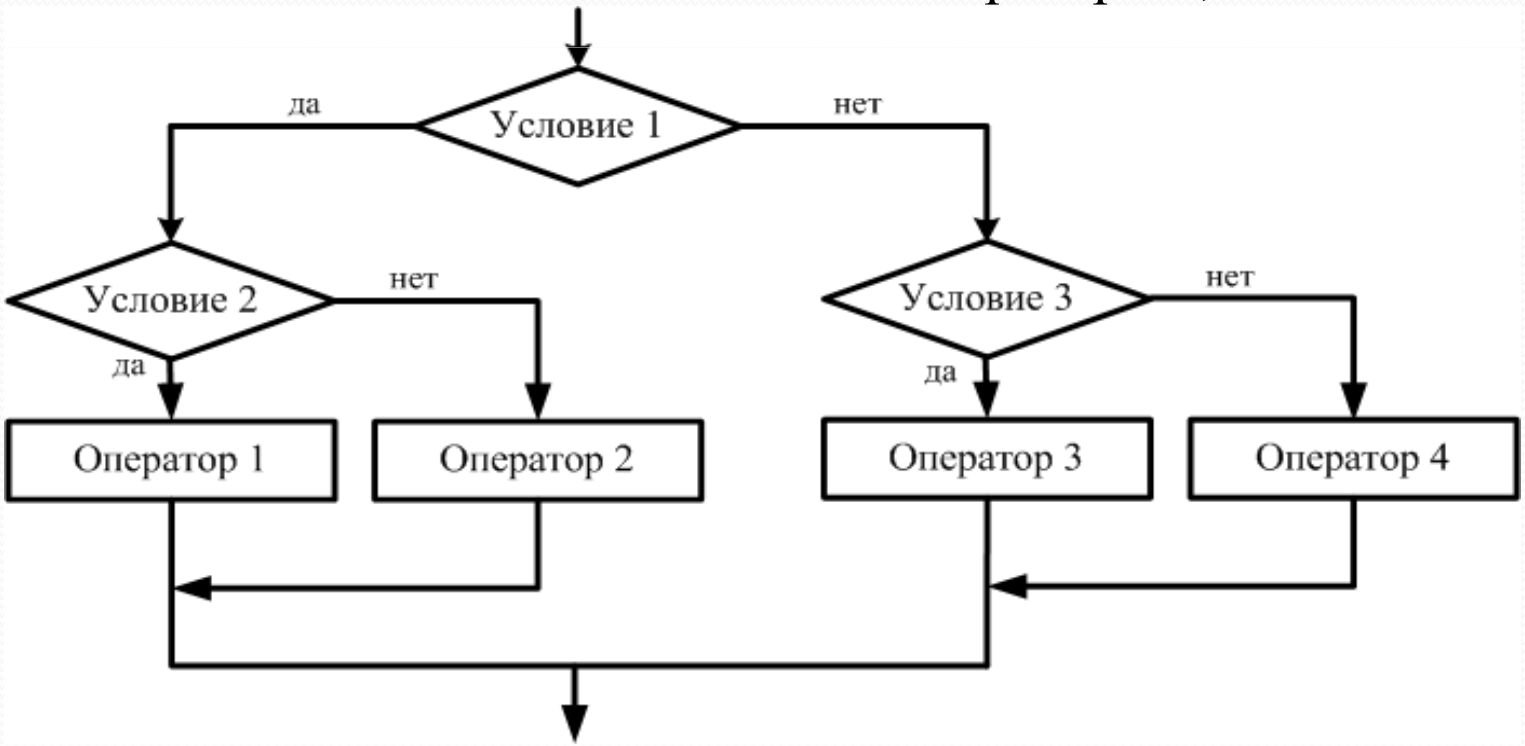
if < Условие1 > **then** < Оператор 1 >;

if < Условие 2> **then** < Оператор 2 >;

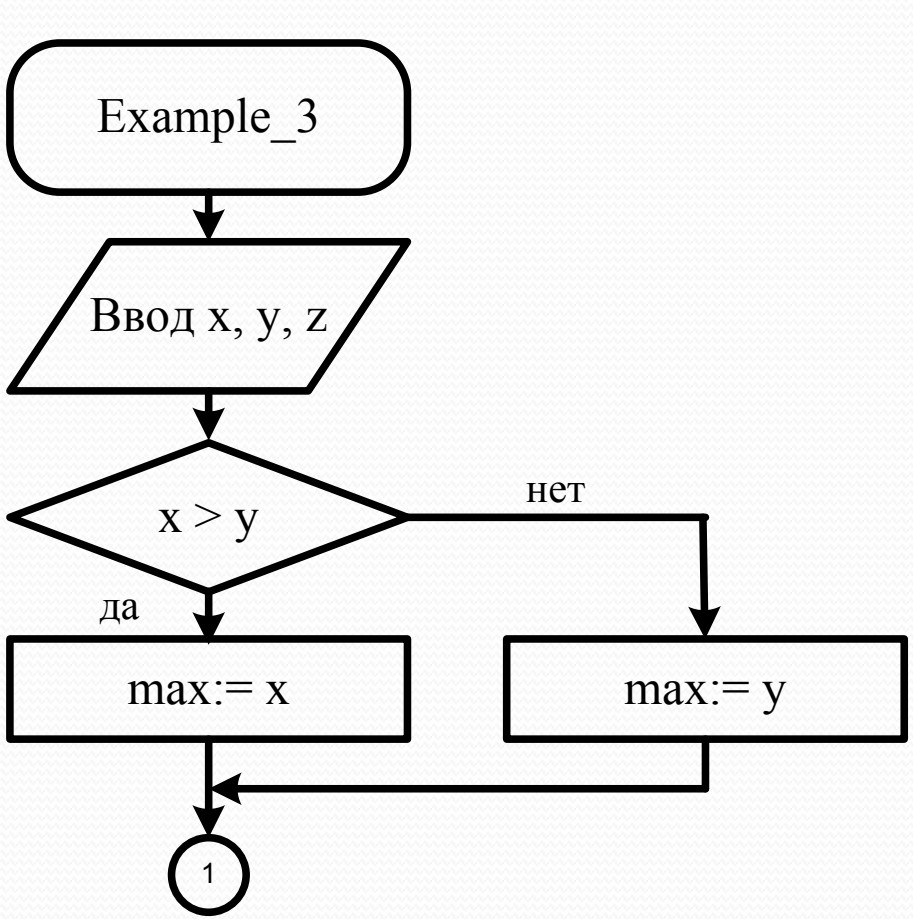


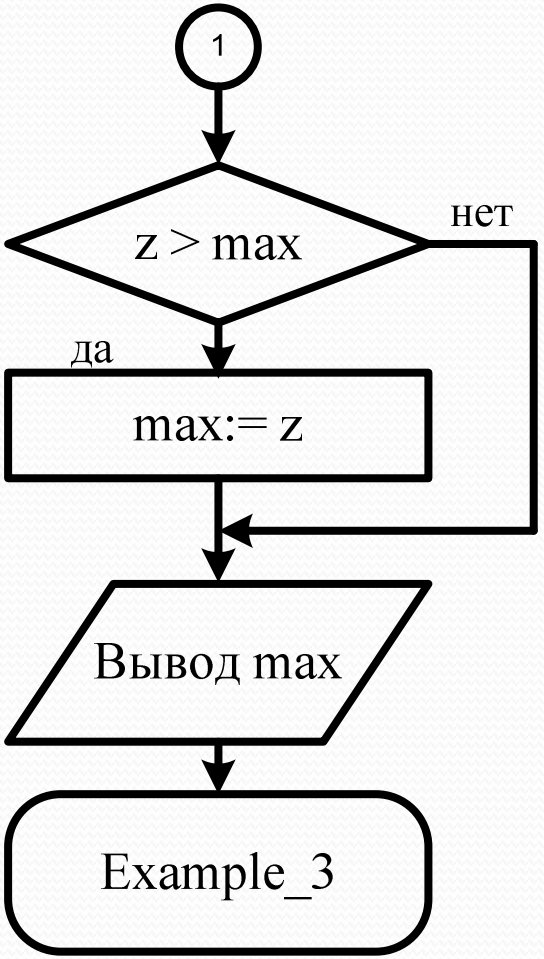
Вложенный условный оператор

IF <условие1> **THEN**
IF <условие2> **THEN** <оператор1>
ELSE <оператор2>
ELSE
IF <условие3> **THEN** <оператор3>
ELSE <оператор4> ;



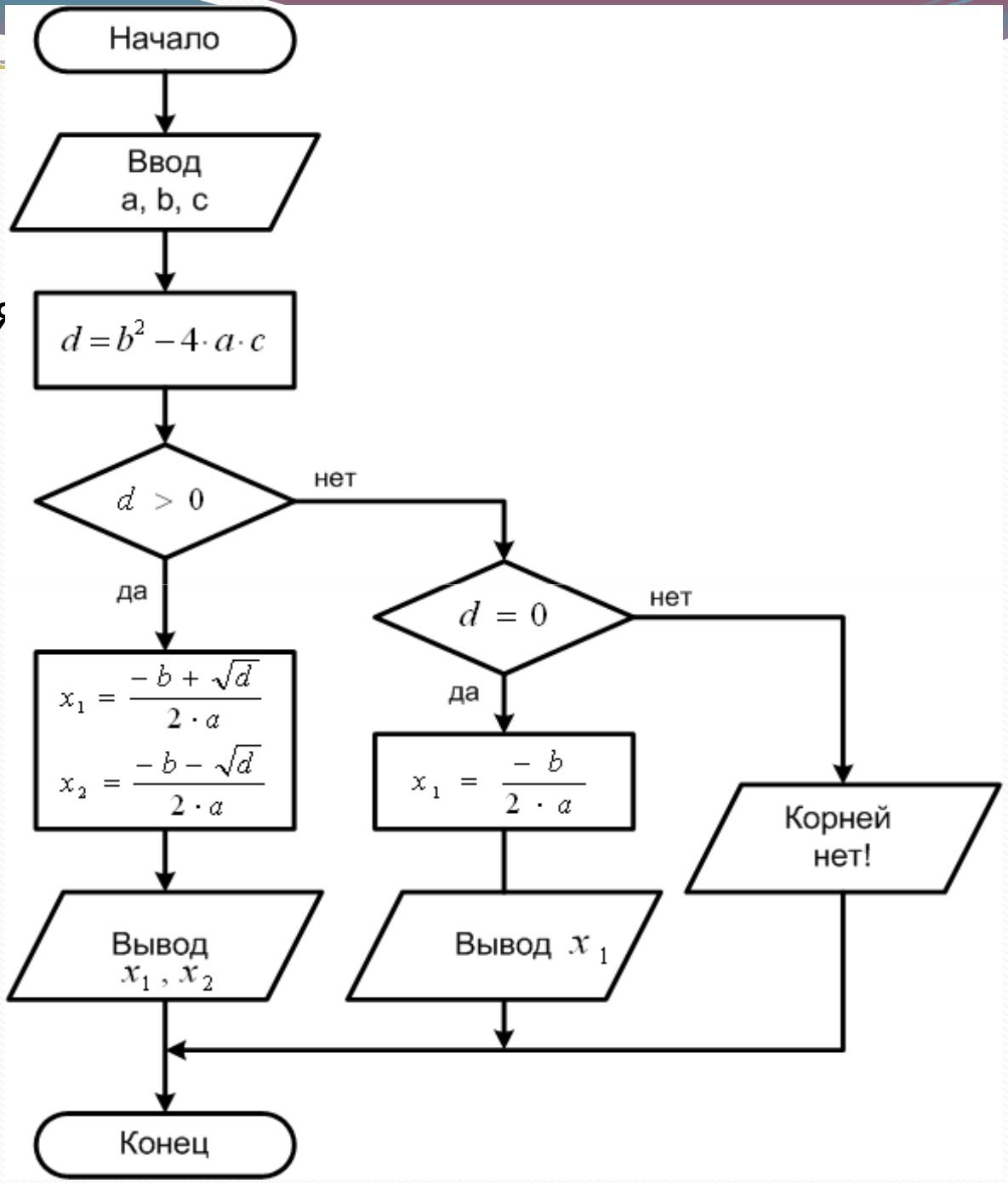
Пример №1 – Даны три вещественных числа. Найти и вывести на экран максимальное из них.





```
Program Example_1;
Uses crt;          {или wincrt}
var x,y,z,max: real;
begin
clrscr;
writeln('Программа нахождения максимума');
write('Введите 3 вещественных числа через пробел:');
readln(x,y,z);
if x>y then max:=x
        else max:=y;
if z>max then max:=z;
writeln('Максимальное из них ',max:4:2);
writeln('Программа завершена. Нажмите Enter.');
readln
end.
```

Пример №2 – Найти все возможные корни квадратного уравнения вида $ax^2+bx+c=0$. Значения *a, b, c* вводятся с клавиатуры.

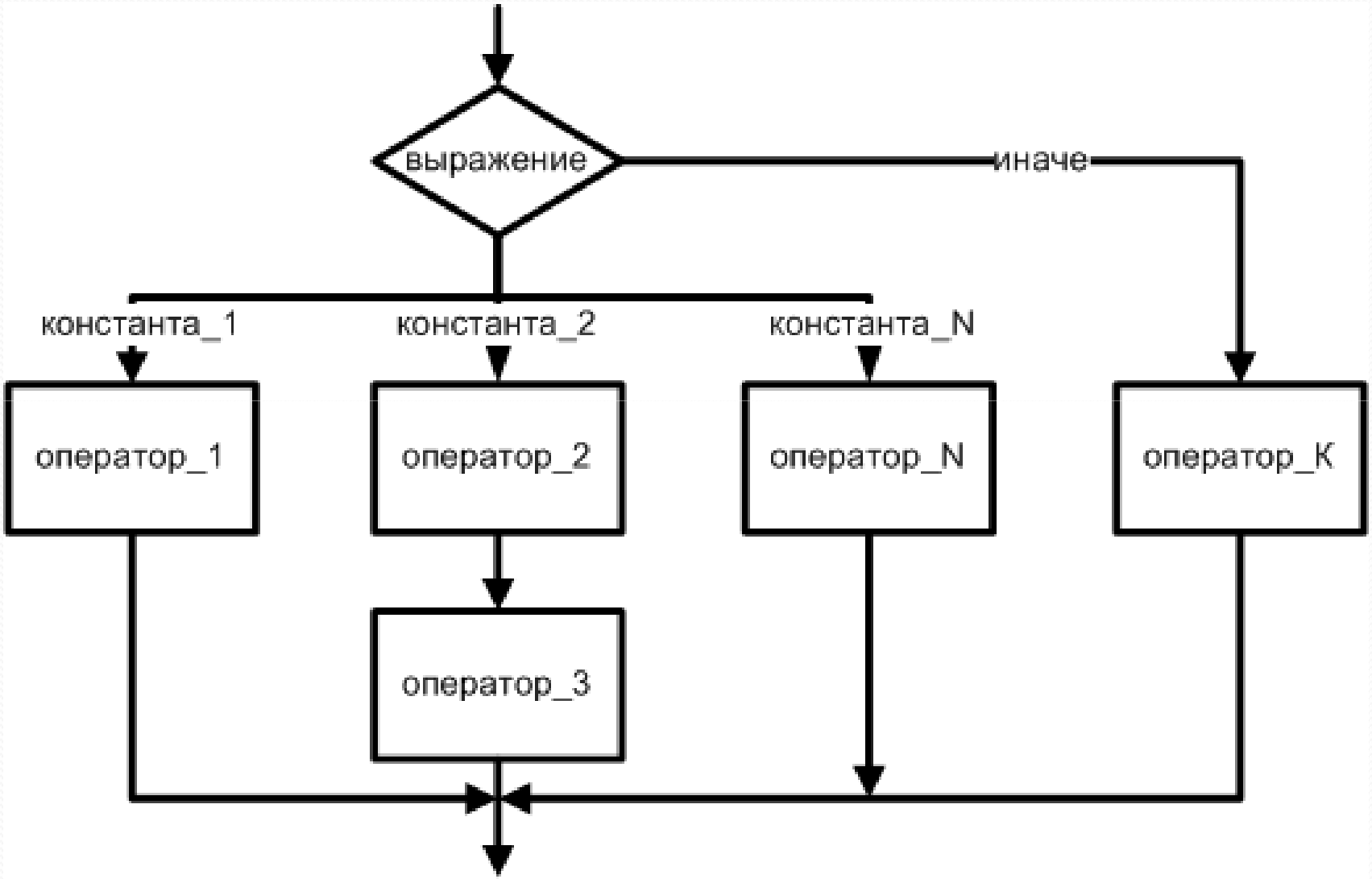


```
Program Example_2;
Uses crt; {или wincrt}
var  a, b, c, d, x1, x2:  real;
begin
clrscr;
Write ('Введите 3 вещественных числа через пробел:');
Readln (a, b, c);
D:=sqr(b)-4*a*c;
If  D>0 then
Begin
X1:=(-b-sqrt(D))/(2*a);   X2:=(-b+sqrt(D))/(2*a);
Writeln( 'корни уравнения X1= ' , X1,   'X2=', X2)
End  else
If  D=0  then
Begin
X1:=(-b)/(2*a);
Writeln( 'корни уравнения X1= ' , X1)
End
Else writeln  ('Действительных корней нет');
writeln('Программа завершена. Нажмите Enter.');
readln
end.
```

Условный оператор CASE (оператор выбора, оператор варианта)

Оператор выбора ***case*** используется, когда необходимо выбрать вариант направления расчетов не из двух, а из большого числа вариантов**.**

```
case <выражение> of
  <константа_1> : <оператор_1> ;
  <константа_2> : begin
                    <оператор_2 >;
                    <оператор_3 >;
                  end;
        .       .       .       .
  <константа_N> : <оператор_N >
   else <оператор_K> ;
end;
```


выражение
иначе
константа_1
константа_2
константа_N
оператор_1
оператор_2
оператор_N
оператор_K
оператор_3

Особенности:

- после case может быть имя переменной или арифметическое выражение целого типа (integer)

```
case i+3 of
 1: begin a := b; end;
 2: begin a := c; end;
end;
```

или символьного типа (char)

```
var c: char;
...
case c of
 'а': writeln('Антилопа');
 'б': writeln('Барсук')
 else writeln('Не знаю');
end;
```

Особенности:

- если нужно выполнить только один оператор, слова begin и end можно не писать

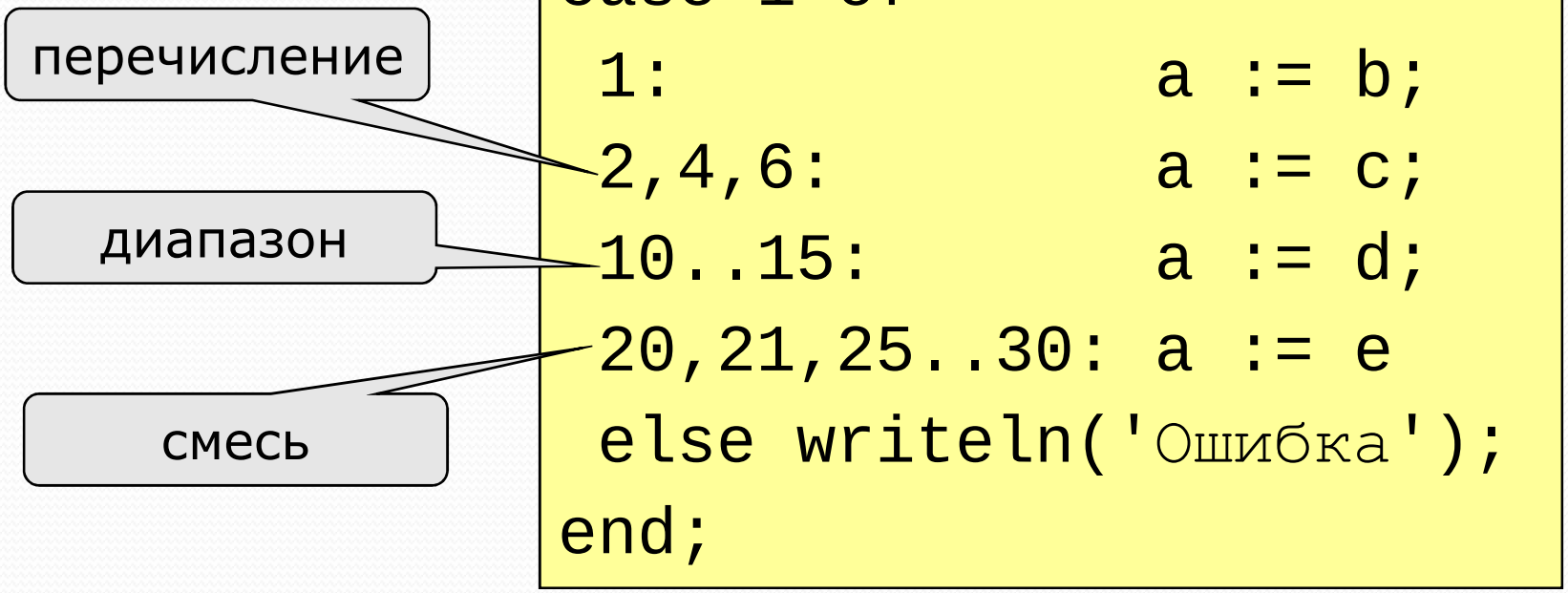
```
case i+3 of
 1: a := b;
 2: a := c;
end;
```

- нельзя ставить два одинаковых значения

```
case i+3 of
 1: a := b;
 1: a := c;
end;
```

Особенности:

- значения, при которых выполняются одинаковые действия, можно группировать



Пример №3: Ввести номер месяца и вывести количество дней в этом месяце.

Решение: Число дней по месяцам:

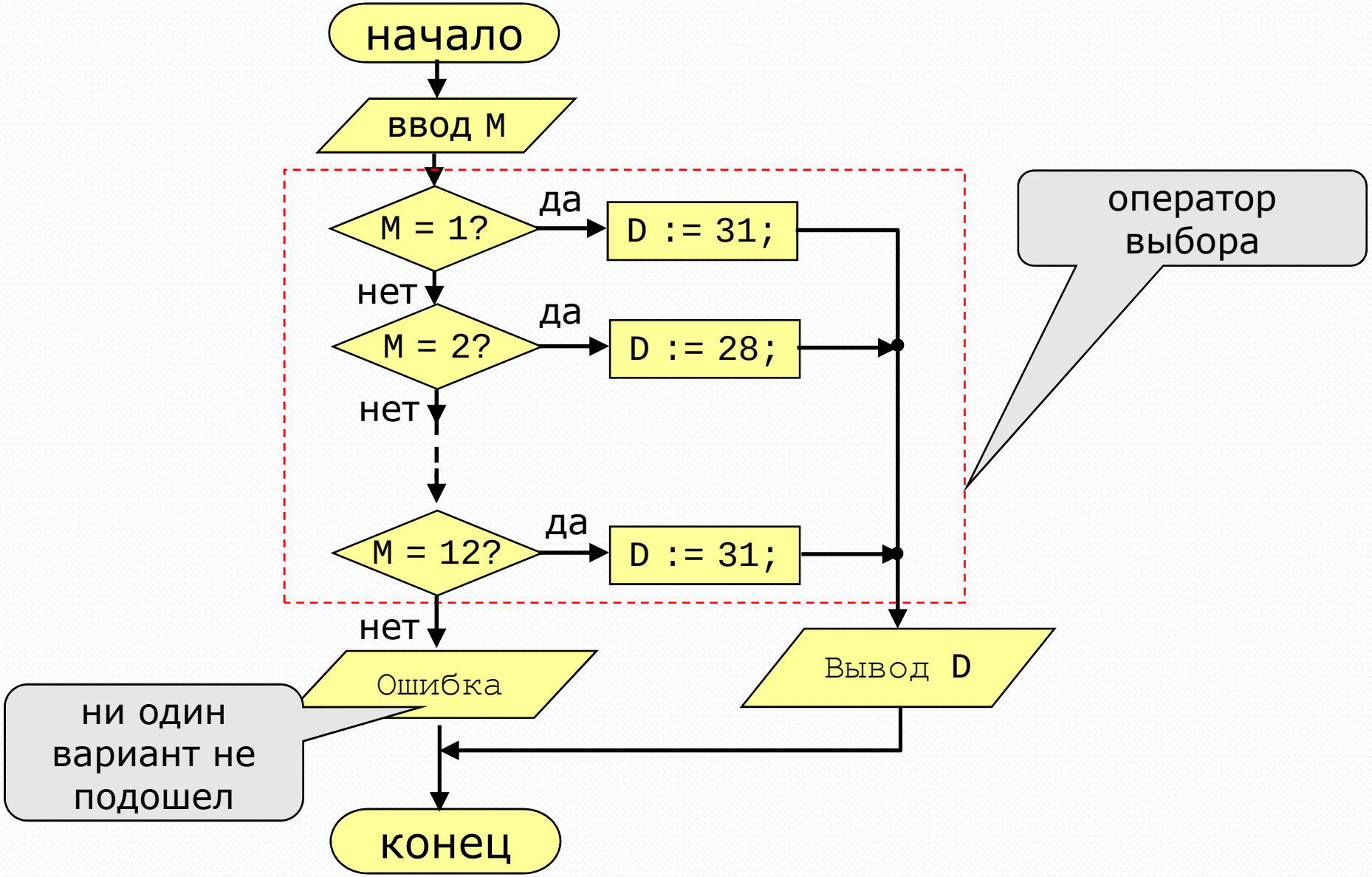
28 дней – 2 (февраль)

30 дней – 4 (апрель), 6 (июнь), 9 (сентябрь), 11 (ноябрь)

31 день – 1 (январь), 3 (март), 5 (май), 7 (июль), 8 (август), 10 (октябрь), 12 (декабрь)

Особенность: Выбор не из двух, а из нескольких вариантов в зависимости от номера месяца.

Алгоритм


начало
ввод M
M = 1?
да
D := 31;
нет
M = 2?
да
D := 28;
нет
M = 12?
да
D := 31;
нет
Ошибка
Вывод D
конец
оператор выбора
ни один вариант не подошел

Программа

```
program Example_3;
var M, D: integer;
begin
   writeln('Введите номер месяца:');
   readln ( M );
   case M of
    2:          begin D := 28; end;
    4,6,9,11: begin D := 30; end;
    1,3,5,7,8,10,12: D := 31
    else                 D := -1;
   end;
   if D > 0 then
      writeln('В этом месяце ', D, ' дней.')
   else
      writeln('Неверный номер месяца');
readln
end.
```

ни один вариант не подошел

Что неправильно?

```
case a of
 2: begin b := a;
 4: b := c;
end;
```

```
case a of
 2: b := a;
 4: b := c
end;
```

```
case a of
 2..5: b := a;
 4: b := c;
end;
```

```
case a of
 0..2: b := a;
 3..6: b := c;
end;
```

```
case a+c/2 of
 2: b := a;
 4: b := c;
end;
```

```
begin
case a of
 2: b := a; d := 0; end;
 4: b := c;
end;
```